

André M. Santamaría Regal

DNI: 36117001D

## Enunciado

1.- Crea un archivo mapper.py que recorra el "Whatsapp\_español.txt" y para cada letra guarde en "salida\_mapper.txt" una pareja de clave-valor donde la clave es la letra y el valor sea 1 indicando que es una aparición. Nuestro mapper no debería distinguir entre mayúsculas y minúsculas ni letras acentuadas.

Por ejemplo, "U", "ü", "ú",... son la misma letra. Los números, espacios en blanco, signos de puntuación, comillas y similares no deben ser tenidos en cuenta, solamente queremos las letras del alfabeto.

Como respuesta a esta pregunta muestra tu código comentado.

```
mapper.py U X
Curso de IA & Big Data > Big data aplicado > Practica Contador letras con MapReduce > mapper.py > ...
1  #!/usr/bin/python3
2  import re
3  from unicodedata import normalize
4  with open("datos/Whatsapp_castellano.txt", "r", encoding="utf8") as archivo_entrada:
5      texto = archivo_entrada.read()
6
7      texto = texto.lower()
8  # NFD y eliminar diacríticos visto en internet la expresion regular
9      texto = re.sub(
10         r"([\u0300-\u036f]|n(?:!\u0303(?:![\u0300-\u036f])))[\u0300-\u036f]+", r"\1",
11         normalize("NFD", texto), 0, re.I
12     )
13
14 texto = normalize('NFC', texto)
15 #Con esto limpiamos el texto de espacios y caracteres que no son letras
16 textolimpio = re.sub(r'^a-zA-Z', '', texto)
17 resultado = [(letra, 1) for letra in textolimpio]
18
19 def escribir_a_archivo(nombre_archivo, lista_tuplas):
20     with open(nombre_archivo, 'w') as archivo:
21         for clave, valor in lista_tuplas:
22             archivo.write(f"{clave}: {valor}\n")
23
24 nombre_archivo = "salida_mapper.txt"
25 escribir_a_archivo(nombre_archivo, resultado)
```



2.- Aplica el archivo Python “ordenar.py” al archivo “salida\_mapper.txt” para ordenar los resultados en un nuevo archivo “entrada\_reducer.txt”.  
Muestra la salida del comando “head -n 20 entrada\_reducer.txt” para ver las 20 primeras líneas del resultado.

3.- Crea un archivo llamado reducer.py que procese el “entrada\_reducer.txt” y nos devuelva el número de apariciones de cada letra de nuestro texto en un archivo llamado “salida\_reducer.txt”. Como respuesta muestra tu código comentado.

```
mapper.py U • ordenar.py U reducer.py U • reducerTry.py U X
Curso de IA & Big Data > Big data aplicado > Practica Contador letras con MapReduce > reducerTry.py > main

1 import re
2 from collections import Counter
3
4 def limpiar_texto(texto):
5     #Limpia el texto eliminando cualquier carácter que no sea una letra.
6     texto_limpio = re.sub(r'^a-zA-Z', '', texto)
7     return texto_limpio
8
9 def contar_letras(texto):
10    #Cuenta el número de apariciones de cada letra en el texto.
11    texto_limpio = limpiar_texto(texto)
12    conteo = Counter(texto_limpio)
13    return conteo
14
15 def leer_archivo(nombre_archivo):
16    #Lee el contenido de un archivo y lo devuelve como una cadena de texto.
17    with open(nombre_archivo, 'r') as archivo:
18        contenido = archivo.read()
19    return contenido
20
21 def escribir_archivo(nombre_archivo, conteo):
22    #Escribe el resultado de contar letras en un archivo de texto.
23    with open(nombre_archivo, 'w') as archivo:
24        for letra, cantidad in sorted(conteo.items()):
25            archivo.write(f"{letra}: {cantidad}\n")
26
27 def main():
28    # Leer el contenido del archivo de entrada
29    texto = leer_archivo('entrada_reducer.txt')
30
31    # Contar las letras en el texto
32    conteo = contar_letras(texto)
33
34    # Escribir el resultado en el archivo de salida
35    escribir_archivo('salida_reducer.txt', conteo)
36
37 # Ejecutar la función principal
38 if __name__ == "__main__":
39     main()
```

4.- Prueba tu mapper y reducer (aplicando ordenar.py como fase intermedia) con el resto de los archivos de prueba y elabora una tabla con las apariciones de cada letra en los distintos idiomas. Devuelve el resultado de cada trabajo mapreduce donde se vea la cantidad de veces que aparece cada letra del alfabeto y una la tabla con los resultados finales de los 4 idiomas.

LETRA	CASTELLANO	FRANCES	INGLES	FILIPINO
a	3332	2289	2106	5591
b	303	206	372	575
c	1792	1270	1023	236
d	1660	1289	892	393
e	3819	5150	2931	895
f	216	302	516	63
g	239	307	415	2564
h	158	144	759	337
i	2691	2713	2102	2097
j	124	33	26	0
k	3	6	64	822
l	1235	1320	921	615
m	687	676	541	1631
n	2251	2503	1712	3010
o	3194	2520	2337	1523
p	986	1057	770	1189
q	214	222	14	2
r	2343	2257	2194	967
s	2980	3172	1838	1451
t	1660	2700	2388	11258
u	1370	1963	1149	766
v	344	658	376	87
w	74	78	403	177
x	66	147	67	1
y	209	82	635	783
z	85	109	16	11